UNIVERSITY OF
Southampton

The Semantic Web at 20

Two decades of developing linked data applications

Dr Nicholas Gibbins – nmg@ecs.soton.ac.uk

> I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A ‘Semantic Web’, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines.

T. Berners-Lee (1999) *Weaving the Web*. San Francisco, CA: Harper

Semantic Web technologies in one slide

Resource Description Framework

- Triple as unit of representation
- Typed relation between entities
- subject-predicate-object
- URIs to identify everyhing

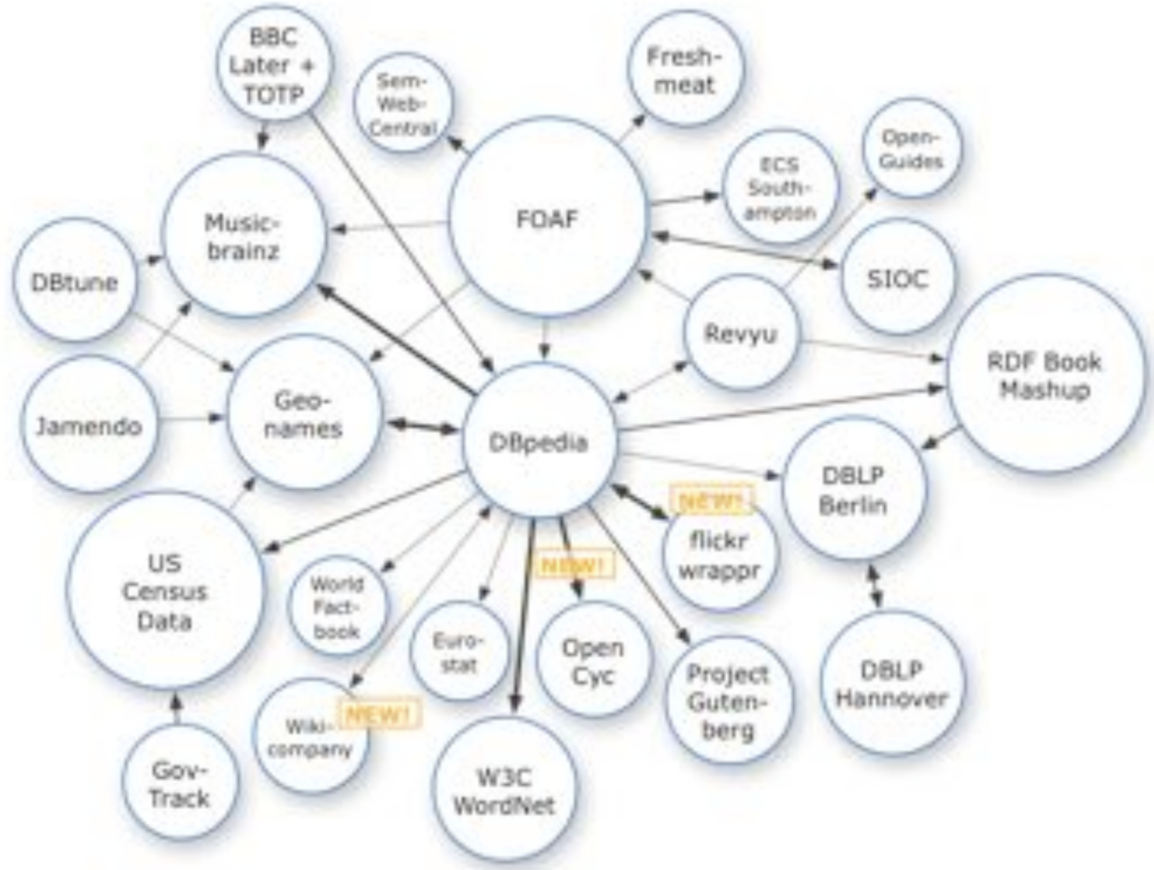
RDF Schema and
OWL (the Web Ontology Language)

- Used to define vocabularies/ontologies

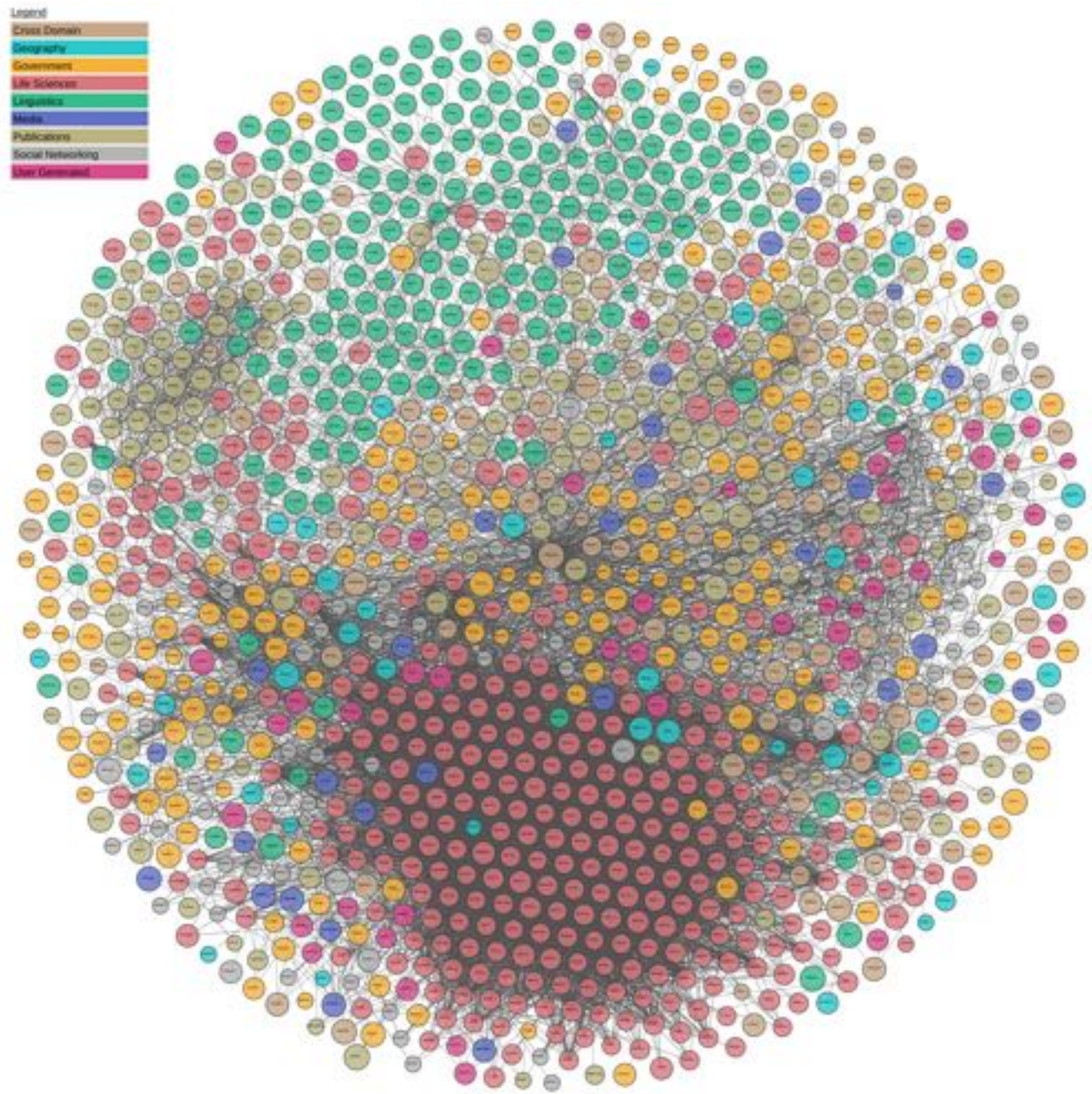
SPARQL Protocol and RDF Query Language

- SQL-like query language for RDF

Web of Data c. 2007



Web of Data c. 2019



The Linked Open Data Cloud from lod-cloud.net

Nine Lessons from/for the Semantic Web

Lesson #1: Beware of the hype

Some early publications raised unrealistic expectations…

The Semantic Web is not a silver bullet

- Not good for quantitative data
- Not good for reasoning with uncertainty

Need to play to the Semantic Web’s strengths:

- Representing qualitative data
- Ad hoc combination of heterogeneous data
- Interoperability

Lesson #2: A little semantics goes a long way

Reasoning at scale with expressive languages is expensive

Many applications require only lightweight reasoning at runtime

Early SW standardisation was dominated by expressive ontology languages

- OWL was overcomplicated by the existence of RDFS (and vice versa)
- Noone stopped to ask the users what they wanted!

Play to the strengths of the technologies

Lesson #3: It’s the triples, stupid

Early developments on the SW suffered from the contemporary enthusiasm for XML

- W3C pushed for the use of XML as a standard serialization format
- RDF has **never** been a well-behaved XML format

RDF has an embarrassing number of “standard” formats

- RDF/XML, N3, Ntriples, Turtle, RDFa, JSON-LD...
- Advocates/detractors for all of the above

All of this is largely irrelevant; the triple-based RDF model is what matters

Need to be format-agnostic

Lesson #4: The myth of “build it, and they will come”

Not enough to put your linked data online and expect people to use it

Early enthusiasm for the use of SW for open government data has been tempered by lack of users

- Complex technologies – high barrier to adoption and use
- Publishing semantic open data has costs
- Hard to sustain without seeing any benefits

Need to understand **how** and **why** users will use your data

Need to support your user community in using your data

Need to be sustainable

Lesson #5: The Semantic Web is part of the Web

TimBL’s original vision for the Semantic Web was explicit about links between data

Somehow, this was lost for much of the first decade of SW development

- Concentration on representation and query languages – OWL, SPARQL
- The last gasp of Good Old-Fashioned Artificial Intelligence

We use URIs to identify SW resources

- What do they **mean**?
- What can you **do** with them?

Need to follow modern best practices for publishing (easier said than done)

Lesson #6: It matters how you publish

It’s not enough to just put your linked data online

Publish all in one file?

- Easy to publish, arduous for consumers who don’t want all of it

Publish broken down into many small files?

- More effort for publisher, arduous for consumers who **do** want all of it

Publish through a query interface (SPARQL)

- Easy for the consumer, potentially very expensive for the publisher

Need to consider how your users will want to interact with your data

Lesson #7: Context matters

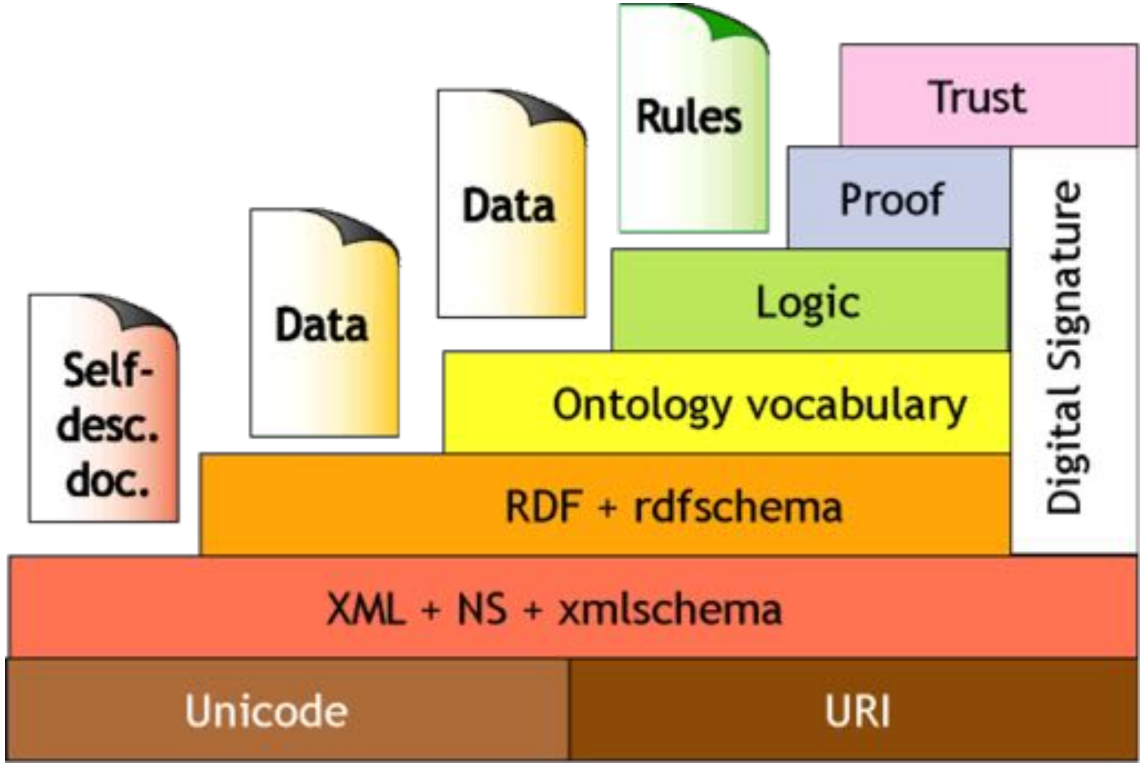
Early talks from TimBL introduced the notion of trust on the SW

- Who said something (digital signatures)
- What is the evidence that what they said was true (proof)

Why should users use/trust your data?

Need to publish data with metadata

- Who published it?
- When was it published? (etc)
- Where did the data come from? (provenance and PROV-O ontology)



Lesson #8: “Standard” is not “Mature”

Many SW recommendations from W3C

- Not all have mature/dependable implementations
- Not all are widely used

Chicken and egg problem!

- No point in adopting a technology without implementation
- No point in implementing a technology without users

Need to make pragmatic decisions when choosing technologies

- Do you have access to robust implementations?
- Do your users have access to robust implementations?

Lesson #9: Eat your own dog food

In the SW community, we've not always practiced what we preach...

- We're very good at producing/promoting technologies that we don't use
- We're very good at producing/promoting data that we don't use

How can we expect users to put up with what we don't/won't?

Need to make sure that we are active and critical consumers of what we make

Future Directions

Future Directions

The foundations of the SW are fixed: RDF, OWL, SPARQL

Some technologies appear to be a dead end
(Rule Interchange Format, Semantic Web Services, etc)

Recent standards work at W3C:

- Making the Semantic Web part of the Web (at last!)
  - Linked Data Platform – robust approaches to publishing linked data
- Representing the context of published data
  - PROV-O and derived ontologies
- Supporting the use of SW technologies in specific domains
  - Range of domain ontology work
  - (but see also schema.org)